UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/915,509	07/26/2001	Cary Lee Bates	ROC920010127US1	6761

7590 06/03/2005

Gero G. McClellan
Thomason, Moser & Patterson, L.L.P.
Suite 1500
3040 Post Oak Boulevard
Houston, TX 77056-6582

EXAMINER	
WOOD, WILLIAM H	

ART UNIT	PAPER NUMBER
2193	

DATE MAILED: 06/03/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/915,509	BATES ET AL.	
	Examiner	Art Unit	
	William H. Wood	2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on 22 November 2004.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 1-29 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 1-29 is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

TR

DETAILED ACTION

Claims 1-29 are pending and have been examined.

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. Claims 1-29 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Borland**'s, "Turbo C++, User's Guide" in view of **Kesselman** et al. (6,678,884).

Claim 1

Borland disclosed a method of determining variables to update in a debugging environment *(page 217-219; page 227-229; and page 237-241)*, the method comprising executing at least one of a first task when a run command is received *(page 228)* and a second task when a set step command for a statement is received *(page 224-225)*, wherein:

(a) the first task comprises determining a first kill variables set comprising variables being monitored by the debugging environment whose values may become out of sync with the corresponding values of those monitored variables while being used by a program being debugged during the execution of a program from a

particular point of the program to a breakpoint, wherein the breakpoint can be encountered during execution of the program from the particular point *(page 228, first through third paragraphs; and pages 237-241)*; and

(b) the second task comprises determining a second kill variables set comprising variables being monitored by the debugging environment whose values may become out of sync with the corresponding values of those monitored variables while being used by a program being debugged by executing of the statement *(page 224-225; page 27, table; and page 237, fourth paragraph)*.

Borland did not explicitly state *first kill variables set comprising only those variables which may be affected by the execution of a program from a particular point ... to a breakpoint*; and *second kill variables set comprising only those variables which may be affected by execution of the statement*. **Kesselman** demonstrated that it was known at the time of invention to utilize code point reaching variable sets (column 6, line 59 to column 7, line 12). It would have been obvious to one of ordinary skill in the art at the time of invention to implement the debugging system of **Borland** with determined variable sets identifying variables which reach or are affected to a point in code (by executing a portion of code to a breakpoint or stepping through code) as found in **Kesselman**'s teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to provide a user with variables and information on variables, which a user expects, regardless of optimizations (**Kesselman**: column 1, lines 54-57). Additionally, this implementation would have

been obvious because one of ordinary skill in the art would be motivated to provide a clear and non-confusing display of user-desired information (selected by the users or for the user automatically), or in other words non-tedious set watches (supported by, **Borland**: page 237, "Monitoring your program by setting watches", third paragraph).

Claim 2

Borland and **Kesselman** disclosed the method of claim 1, wherein determining the first kill variables set comprises referencing a kill set of a statement, wherein the kill set of the statement contains variables affected by the statement *(**Kesselman**: column 3, lines 35-39)*.

Claim 3

Borland and **Kesselman** disclosed the method of claim 1, further comprising, prior to determining the first kill set, determining whether any breakpoints may be encountered during subsequent continuing execution from the particular point *(**Kesselman**: column 3, lines 25-32)*.

Claim 4

Borland and **Kesselman** disclosed the method of claim 1, further comprising, where the second task is executed:

- setting a step operation for the statement *(**Borland**: pages 224-225)*;
- executing the step operation *(**Borland**: pages 224-225)*; and

- updating on a user interface only the variables contained in the second kill variables set *(**Borland**: pages 237-241)*.

Claim 5

Borland and **Kesselman** disclosed the method of claim 1, wherein the first task is performed for a plurality of breakpoints, each of which may be encountered during execution of the program from the particular point in the program, whereby an instance of the first kill variables set is provided for each of the plurality of breakpoints *(**Borland**: page 227-228, multiple breakpoints; **Kesselman**: column 6, line 59 to column 7, line 11, analyze entire path, multiple breakpoints and all)*.

Claim 6

Borland and **Kesselman** disclosed the method of claim 5, further comprising, where the first task is executed:

- executing the run command *(**Borland**: page 27, table)*;
- hitting a particular breakpoint of the plurality of breakpoints *(**Borland**: page 227-229; and **Kesselman**: column 3, lines 25-34)*; and
- updating on a user interface only the variables contained in a respective first kill variables set associated with the particular breakpoint *(**Borland**: page 227-241, watches; and **Kesselman**: column 3, lines 25-34)*.

Claim 7

Borland and **Kesselman** disclosed the method of claim 1, further comprising, where the first task is executed:

- executing the run command *(**Borland**: page 27, table)*;
- hitting the breakpoint *(**Borland**: page 227-229; and **Kesselman**: column 3, lines 25-34)*; and
- updating on a user interface only the variables contained in the first kill variables set *(**Borland**: page 227-241, watches; and **Kesselman**: column 3, lines 25-34)*.

Claim 8

Borland and **Kesselman** disclosed the method of claim 7, wherein the updating is only performed if the first task has completed execution *(clearly the variables can only be updated once they are identified)*.

Claim 9

Borland and **Kesselman** disclosed the method of claim 1, wherein determining the first kill variables set comprises:

- if a node containing the particular point of the program contains at least one breakpoint, determining kill variables of each statement of the node from the particular point up to a first encountered breakpoint, wherein the first encountered breakpoint is a first breakpoint encountered when traversing the

node from the particular point and wherein the kill variables of a respective statement are those variables which may be affected by execution of the respective statement *(**Kesselman**: column 6, line 59 to column 7, line 11)*; and then

- generating the first kill variables set by associating all determined local kill variables with the first encountered breakpoint *(**Kesselman**: column 6, line 59 to column 7, line 11)*.

Claim 10

Borland and **Kesselman** disclosed the method of claim 9, further comprising updating on a user interface only the variables contained in the first kill variables set *(**Borland**: page 227-241, watches; and **Kesselman**: column 3, lines 25-34)*.

Claim 11

Borland and **Kesselman** disclosed the method of claim 9, wherein determining the first kill variables set further comprises, if the node does not contain at least one other breakpoint:

- beginning with a root node, marking each node of a control flow graph which may be exited and reentered during execution of the program from the particular point *(**Kesselman**: column 3, lines 13-24)*;
- for each marked node, marking all variables of the control flow graph as a kill variable *(**Kesselman**: column 5, lines 48-49)*; and

- beginning with the node containing the particular point of the program, generating a list of unmarked nodes which may be reached during execution from the particular point *(**Kesselman**: column 6, line 59 to column 7, line 11)*.

Claim 12

Borland and **Kesselman** disclosed the method of claim 11, wherein generating the list comprises:

- traversing the control flow graph from the node containing the particular point of the program to each subsequent node by following program control flow defined by arcs *(**Kesselman**: column 2, line 48 to column 3, line 24; and column 6, line 59 to column 7, line 11)*;
- adding each encountered node to the list if the encountered node is not marked and is not already in the list *(**Kesselman**: column 7, lines 3-11)*;
- determining whether the encountered node contains a breakpoint *(**Kesselman**: column 6, line 59 to column 7, line 11)*; and
- if so, terminating a traversal along a current traversal path *(**Kesselman**: column 6, line 59 to column 7, line 11)*.

Claim 13

Borland and **Kesselman** disclosed the method of claim 11, further comprising. propagating, to each unmarked node in the list, a propagated kill set from a preceding unmarked node in the control flow graph, wherein the propagated kill set contains a sum

of all kill variables associated with all statements of the preceding unmarked node and all propagated kill variables propagated to the preceding unmarked node *(**Kesselman**: column 6, line 59 to column 7, line 11)*.

Claims 14-26

The limitations of computer readable medium claims 14-16 correspond to the limitations of method claims 1-13 and as such are rejected in the same manner.

Claim 27

Borland disclosed a computer readable medium containing a program which, when executed by a processor, performs operation for determining variables to update in a debugging environment *(page 217-219; page 237-241)*, the operations comprising:

- executing a task when a run command is received, wherein the task comprises determining a kill variables set comprising variables being monitored by the debugging environment whose values become out of sync with the corresponding values of those monitored variables while being used by a program being debugged during the execution of a program from, and including, a particular statement of the program to a breakpoint that can be encountered during executing of the program from the particular point *(page 228, first through third paragraphs; and pages 237-241)*; and
- if the task has completed execution when the breakpoint is encountered during execution of the program, updating on a user interface the variables

contained in the kill variables set *(**Borland**: page 227-241, watches; and **Kesselman**: column 3, lines 25-34).*

Borland did not explicitly state *kill variables set comprising only those variables which may be affected by the execution of a program from a particular point.* **Kesselman** demonstrated that it was known at the time of invention to utilize code point reaching variable sets (column 6, line 59 to column 7, line 12). It would have been obvious to one of ordinary skill in the art at the time of invention to implement the debugging system of **Borland** with determined variable sets identifying variables which reach or are affected to a point in code (by executing a portion of code to a breakpoint or stepping through code) as found in **Kesselman**'s teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to provide a user with variables and information on variables, which a user expects, regardless of optimizations (**Kesselman**: column 1, lines 54-57). Additionally, this implementation would have been obvious because one of ordinary skill in the art would be motivated to provide a clear and non-confusing display of user-desired information (selected by the users or for the user automatically), or in other words non-tedious set watches (supported by, **Borland**: page 237, "Monitoring your program by setting watches", third paragraph).

Claim 28

Borland and **Kesselman** disclosed the computer readable medium of claim 27, wherein determining the kill variables set comprises:

- beginning with a root node, marking each node of a control flow graph which may be exited and reentered during execution of the program from the particular statement *(**Kesselman**: column 3, lines 13-24)*;
- for each marked node, marking all variables of the control flow graph as a kill variable *(**Kesselman**: column 5, lines 48-49)*; and
- beginning with the node containing the particular statement, generating a list of unmarked nodes which may be reached during execution from the particular statement *(**Kesselman**: column 6, line 59 to column 7, line 11)*.

Claim 29

Borland and **Kesselman** disclosed the computer readable medium of claim 28, propagating, to each unmarked node in the list, a propagated kill set from a preceding unmarked node in the control flow graph, wherein the propagated kill set contains a sum of all kill variables associated with all statements of the preceding unmarked node and all propagated kill variables propagated to the preceding unmarked node *(**Kesselman**: column 6, line 59 to column 7, line 11)*.

Response to Arguments

3. Applicant's arguments filed 22 November 2004 have been fully considered but they are not persuasive. Applicant argues no existing variables sets in the cited prior art (response, page 11, middle paragraph). With this assertion, respectful disagreement is found. Note sets of variables being monitored on at least pages 228 and 237 of

Borland. Thus, kill variable sets that are monitored for out of sync behavior are found as indicated. Therefore, the rejections are maintained.

Conclusion

4. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Correspondence Information

Any inquiry concerning this communication or earlier communications from the examiner should be directed to William H. Wood whose telephone number is (571)-272-3736. The examiner can normally be reached 9:00am - 5:30pm Monday thru Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (571)-272-3719. The fax phone numbers for the organization where this application or proceeding is assigned are (703)872-9306 for regular communications.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703)305-3900.

William H. Wood
May 31, 2005

ANIL KHATRI
PRIMARY EXAMINER